

Centre de Physique Théorique - CNRS - Luminy, Case 907
F-13288 Marseille Cedex 9 - France
Unité Propre de Recherche 7061

TEXTURE SEGMENTATION BY LOCAL BI-ORTHOGONAL DECOMPOSITION

J.A. DENTE^{*}, R. LIMA[†] and R. VILELA MENDES^{*}

Abstract

We investigate the ability of a local bi-orthogonal decomposition to build texture segmentation of images. Using the structures associated to the local decomposition of the image independent row and columns we perform a segmentation, where the regions are defined by the property of having a smooth variation of the corresponding entropy. Examples are chosen in texture made and also in real life images. The size of the local analysis is also determined by the properties of the (global) bi-orthogonal decomposition.

Key-Words : texture segmentation, bi-orthogonal decomposition, entropy.

Number of figures : 18

January 1994
CPT-94/P.3010

anonymous ftp or gopher : cpt.univ-mrs.fr

^{*} Laboratório de Mecatrónica Dept. Eng. Electrotécnica, Instituto Superior Técnico,
av. Rovisco Pais 1, 1096 Lisboa Codex Portugal

[†] Centre de Physique Théorique, CNRS - Luminy, Case 907, 13288 Marseille Cedex 9

1. TEXTURE SEGMENTATION AND THE LOCAL BI-ORTHOGONAL DECOMPOSITION

Image segmentation is the partition of a plane image into exclusive regions which, in some sense, are homogeneous. When the purpose of segmentation is to distinguish some object from a background and the brightness of the object and the background are significantly different, segmentation by *gray level thresholding* is a possibility. This works well for example in automated manufacturing processes if the assembly parts are kept dark against a bright background but, in most images of 3-dimensional objects, different illumination levels in different parts of the same object make gray level thresholding a poor discriminating technique.

A more frequently used technique is *edge detection* by convolution of the image with a discrete difference operator, followed by a contour filling algorithm to decide which pixels belong to each one of the segmented regions. Edge detection also faces serious difficulties because factors such as illumination may either hide physical boundaries or, through shadows, cause brightness discontinuities which are not related to any real boundaries.

The reason why segmentation in computer vision is such a difficult problem, as compared with the ease with which the “eye plus brain” system performs this task, is because in the brain an huge amount of information is stored concerning the way the real world looks like. Based on a few external stimuli, like brightness levels and a few contours, the bulk of the segmentation process in the brain is likely to be mostly an exercise in pattern matching of the external stimuli with our “image of the world” data basis. While our computers are not equipped with a data basis of comparable size and complexity as the brain, computers must rely on a refinement in the analysis of the external stimuli part of the process. This means that quantitative characterizations of global and local properties of the image must be developed, which might even have to be finer and more accurate than those performed by the human eye. Only then, might we compensate for the weakness of the data basis in computer vision.

The most difficult of all segmentation problems occurs when different regions of the image cannot be distinguished by gray level nor by sharp boundaries, but only by a difference in texture. Texture refers to the *local* characteristics of the image. A local gray level histogram is a local statistical parameter. However, for the texture, what matters most are the local spatial correlations between pixel intensities. The brain will probably perform the *texture segmentation* task by pattern matching with its data basis but, in the computer, the only alternative is to attempt an objective mathematical characterization of what texture means. Several quantities have been proposed as a measure of texture. For example :

- The *gray level co-occurrence matrix* $P^{[1,2]}$ is a matrix with elements $P(i, j)$ which are the number of pairs of pixels that in some neighborhood have intensities i and j .
- The *local autocorrelation function*^[3]

$$A_I(m, n) = \sum_j \sum_k I(j, k) I(j - m, k - n)$$

where $I(i, j)$ is the image intensity at the point with coordinates (i, j) and the sum is over a small window $-D \leq m, n \leq D$, or quantities derived from $A_I(m, n)$, like the *autocorrelation spread measures*^[4].

- The *number of edges*^[3] in a neighborhood
- The *local Fourier spectrum*^[5,6]
- The *singular value decomposition*^[7] of local texture samples
- The *moments of the gray-level histograms* of small windows
- *Texture primitives* and *grammar rules* to generate the pattern

In this paper we are proposing and testing the idea that the entropy associated to the bi-orthogonal decomposition is an adequate parameter to characterize different textures. The bi-orthogonal decomposition (see Appendix A and Ref. [13] for more details) is a 2-dimensional generalization of the Karhunen-Loève^[8,9] technique which states that a real signal $u(x, y)$ on two variables may be uniquely decomposed in the form

$$u(x, y) = \sum_k \alpha_k \phi_k(x) \Psi_k(y) \quad (1)$$

where $\{\phi_k(x)\}$ and $\{\Psi_k(y)\}$ are both orthonormal sets. The expansion basis is generated by the signal itself and the pairs $\phi_k(x) \Psi_k(y)$ are the independent x, y -structures that compose the image. They encode the full nature of the geometrical 2-dimensional correlations in the image. Two-dimensional spatial correlations between pixel intensities being at the very root of the notion of texture, it is natural to conjecture that the bi-orthogonal decomposition of local blocks of appropriate size is an appropriate tool to characterize textures in an image. We therefore propose the following three-step process for texture segmentation by local bi-orthogonal decomposition (LBOD) :

i) Identify the texture average scale

Compute the Fourier transform of a few randomly chosen lines and columns of the image. Next the algorithm should identify the first peak in the spectrum after the peak around zero (which corresponds to the average pixel intensity and long-range slow variations). In typical images the first large peak away from zero is the lowest texture frequency ω_T . A block size $M \times N$ is then chosen where M and N correspond to the average $1/\omega_T$ along the lines and the columns. Instead of using the Fourier transform

of a set of lines and columns, we may use the Fourier transform of one of the modes in a global bi-orthogonal decomposition of the image.

ii) Construct the entropy image

The image is now divided into blocks of size $M \times N$ and the bi-orthogonal entropy (Eq. B.2) of each block is computed. (For a comparison of the bi-orthogonal entropy with other entropy notions see Appendix B). Assigning to each block its entropy value one obtains a *block entropy image*.

iii) Segmentation from the entropy image

The entropy image is smoothed by some standard algorithm and contour tracing from the smoothed entropy image completes the process of texture segmentation by LBOD.

2. EXAMPLES

Before dealing with real world images we tested the algorithm on the image shown in figure 1. This image has several textures which were constructed in such a way that the local (in 8×8 blocks) gray level average is everywhere the same and no boundary lines exist separating the different textures. In this sense this example presents a pure case of segmentation by textures.

The average texture scale was found by computing the Fourier transform of the eigenfunctions ϕ_k of the global bi-orthogonal decomposition. Figure 2 shows the spectrum of ϕ_2 . A large peak may be seen, that corresponds to a block of dimension 8×8 pixels.

The image is then divided into blocks of size 8×8 and the bi-orthogonal entropy of each block is computed to obtain the block entropy image shown in figure 3.

In the entropy image, zones with different textures are well separated by the entropy values. This enables us to use a simple gradient algorithm to find the contours, thus performing the texture segmentation. Figure 4 shows the result of this operation.

To obtain an entropy image with better resolution we might compute the entropy in a neighbourhood of every pixel in the original image. However this procedure is time consuming. It suffices to generate an entropy image using the block entropy for only a smaller number of pixels in the original image. Figures 5a,b show the entropy image evaluated using 8×8 blocks separated by 4 pixels. Figure 6 shows the contours obtained in this case.

If the local entropy actually characterises the local texture, it should not be too sensitive to illumination levels in the image. We have tested this feature by changing the intensity in one half of our test image (Figure 7).

The relative insensitivity of the block entropies to illumination levels is apparent from the entropy and contour images shown in the figures 8 and 9.

In the “pure textures” example described above, texture segmentation by local bi-orthogonal decomposition seems to work efficiently. In real world images, however, we see some difficulties and limitations of the method. Take for example the “bears” image of figure 10.

The first difficulty occurs in the choice of the block size. In a real world image many different texture scales may occur, hence there is no unique block size appropriate for all texture features. Figure 11 shows the spectrum of the global ϕ_2 eigenfunction of the “bears” image. A large dimension is suggested for the dominant block size. This correspond to large areas of the image (the water and the mountain) which are not the main objects in the image. If other eigenfunctions are used other characteristic block sizes are found, as shown in the figures 12 and 13. The low block sizes are associated to low energy levels, but in spite of this their small intensities they are important to define the details of the image, that is the micro-structures that our brain understands.

The effect of the block size is shown in the entropy images of figures 14 and 15, which use blocks of size (18×18) and (4×4) respectively.

Contours are difficult to obtain in a simple way, because the entropy image has many different values as shown in three dimension figures 16 and 17. With a simple gradient plus clipping algorithm one obtains the result show in figure 18.

Appendix A : The bi-orthogonal decomposition

The decomposition into orthogonal modes, of probability theory, is a well known procedure in signal analysis referred to as Karhunen-Loève decomposition^[8,9] or principal component analysis. Given a random vector $x_i \{i = 1, \dots, N\}$, $x_i \in X$, the covariance matrix $Q = [xx^T]$ is diagonalized and the random vector x expressed as

$$x = \sum_{i=1}^N \alpha_i \phi_i \quad (\text{A.1})$$

where ϕ_i are the eigenvectors of Q , i.e. the columns of the matrix A that diagonalizes $Q (Q = A\lambda A^T, \lambda \text{ diagonal})$. The best (mean-square) P -component approximation to the signal x (with $P < N$) is obtained choosing the α_i coefficients associated with the largest P eigenvalues. This property makes the Karhunen-Loève decomposition a standard data compression technique. The Karhunen-Loève technique has been used for image processing^[10–12]. The image is divided into small blocks, each block is treated as a sample of an one-dimensional statistical signal, the labeling of the blocks playing

the role of time variable. The expectation value in the covariance $E[xx^T]$ is then taken over these blocks.

However, in a image, an important part of the relevant information is related to geometrical correlations. They concern the variation of the gray levels along particular directions and define the contour and the shape information content of the image. This suggests the use, for image processing, of a generalization of the Karhunen-Loève technique where bidimensional correlations are explicitly taken into account.

The bi-orthogonal decomposition analyses signals $u(x, y)$ that depend on variables defined in two different spaces ($x \in X$, $y \in Y$). We summarize below the main results concerning the bi-orthogonal decomposition and refer to [13] for more details.

Let the signal $u(x, y)$ be a measurable complex-value function defined on $X \times Y$, where X and Y are either R^n or Z^n or subsets of one of these. The signal defines a linear operator $U : L^2(Y) \rightarrow L^2(X)$ by

$$(U\Psi)(x) = \int_Y u(x, y)\Psi(y)dy \quad \forall \Psi \in L^2(Y) \quad (\text{A.2a})$$

with adjoint operator $U^\dagger : L^2(X) \rightarrow L^2(Y)$

$$(U\phi)(y) = \int_X u^*(x, y)\phi(x)dx \quad \forall \phi \in L^2(X) \quad (\text{A.2b})$$

The analysis of the signal $u(x, y)$ is the spectral analysis of the operator U . In general the spectrum contains continuous and point spectral components. However we will assume that $u \in L^2(X \times Y)$ or that X and Y are compact and u continuous implying that U is a compact operator. Then the spectrum consists of a countable set of isolated points. There is a canonical decomposition of $u(x, y)$ such that

$$u(x, y) = \sum_{k=1}^{\infty} \alpha_k \phi_k(x) \Psi_k^*(y) \quad (\text{A.3})$$

is norm-convergent

$$\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_k \geq \dots > 0, \quad \lim \alpha_k = 0, \quad (\phi_i, \phi_j) = (\Psi_i, \Psi_j) = \delta_{i,j}$$

The functions $\phi_k(x)$ are also eigenfunctions of the operator $L = UU^\dagger$, and the $\Psi_k(y)$ are eigenfunctions of $R = U^\dagger U$. These functions are related by the following equation

$$\phi_k = \alpha_k^{-1} U \Psi_k \quad (\text{A.4})$$

The operators L and R are non-negative operators with kernels $l(x_1, x_2)$ and $r(y_1, y_2)$ that are the X and Y -correlation functions of the signal

$$l(x_1, x_2) = \int_Y u(x_1, y)u^*(x_2, y)dy \quad (\text{A.5})$$

$$r(y_1, y_2) = \int_x u^\star(x, y_1)u(x, y_2)dx \quad (\text{A.6})$$

The eigenvectors ϕ_k and Ψ_k of the L and R operators appear, in the decomposition (A.3) of the signal, intrinsically coupled to the same eigenvalue α_k^2 . The products $\phi_k \Psi_k$ are therefore the independent X, Y structures that compose the signal. This decoupling of structures occurs because, as opposed to other methods of signal analysis (Fourier, wavelets, etc), the functional basis decomposing u is produced by u itself.

From the bi-orthogonal decomposition one may construct several global quantities : The square of the norm of the signal in $L^2(X \times Y)$, which we call the energy, equals the sum of the eigenvalues

$$E(u) = \int_{X \times Y} u(x, y)u^\star(x, y)dxdy = \sum_k \alpha_k^2 \quad (\text{A.7})$$

Similarly one defines X -dependent and Y -dependent energies as

$$E_x(u) = \int_Y u(x, y)u^\star(x, y)dy = \sum_k \alpha_k^2 |\phi_k(x)|^2 \quad (\text{A.8})$$

$$E_y(u) = \int_X u(x, y)u^\star(x, y)dx = \sum_k \alpha_k^2 |\Psi_k(y)|^2 \quad (\text{A.9})$$

The dimension of a signal is defined to be the dimension of the range of U . For the compact case this is the number of non zero eigenvalues α_k^2 .

The ε -dimension of the signal is the number of eigenvalues larger than ε . The size of the eigenvalues is a good characterization of the degree of approximation in the sense that, truncating the U operator to

$$U_p = \sum_k^p \alpha_k \phi_k \Psi_k$$

the norm of the error $\|U - U_p\|$ is smaller than the first neglected eigenvalue. The notion of ε -dimension is useful to characterize noisy signals.

Appendix B : Entropy

The notion of entropy may be used to estimate the information content. It measures the amount of disorder in a system and, in this sense, it is sensitive to the spread of possible states which a system can adopt. For an image the simplest idea is to make these states correspond to the possible values which individual pixels can adopt. Then, the entropy (associated to the gray level histogram) would be given by

$$E = - \sum_{j=0}^{M-1} P(j) \log P(j) \quad (\text{B.1})$$

where $P(j)$ is the probability of pixel value j and M is the number of different values which the pixels can take. Equation (B1) represents the information content of the image only if all pixels are uncorrelated. This is not the case in real world images. Consider, instead of the original image, an image formed by the differences of neighbouring pixels. The original image can be reconstructed from the difference image together with the value of the first pixel. Therefore they contain the same information. However one usually finds that the entropy of the gray level histogram of the “difference image” is smaller than the one for the original image. This occurs because the difference image extracts some of the space correlations existing in the image, hence its entropy is closer to the actual information content of the image.

As explained in Appendix A, the bi-orthogonal decomposition extracts the normal modes of the image fully taking into account the correlations along the two coordinate axis. Therefore we expect that the entropy associated to the weights of the modes in the bi-orthogonal decomposition would be even closer to the actual information content of the image.

Associated to the eigenvalue structure of the bi-orthogonal decomposition we define an entropy by

$$H(u) = - \lim_{N \rightarrow \infty} \frac{1}{\log N} \sum_{k=1}^N p_k \log p_k \quad (\text{B.2})$$

where

$$p_k = \frac{\alpha_k^2}{\sum_k \alpha_k^2} \quad (\text{B.3})$$

and X and Y -entropies by

$$H_x(u) = - \lim_{N \rightarrow \infty} \frac{1}{\log N} \sum_{k=1}^N p_k(x) \log p_k(x) \quad (\text{B.4})$$

$$H_y(u) = - \lim_{N \rightarrow \infty} \frac{1}{\log N} \sum_{k=1}^N p_k(y) \log p_k(y) \quad (\text{B.5})$$

where

$$p_k(x) = \frac{\alpha_k^2 |\phi_k(x)|^2}{\sum_k \alpha_k^2 |\phi_k(x)|^2} \quad ; \quad p_k(y) = \frac{\alpha_k^2 |\Psi_k(y)|^2}{\sum_k \alpha_k^2 |\Psi_k(y)|^2}$$

We have computed the gray level histogram entropy, the entropy of the difference images and the entropy of the bi-orthogonal decomposition for real world images and for our textures test image. For real world images we typically find that the entropy of the difference image is smaller than the gray level histogram entropy. An exception is our textures test image. This is because the local textures lead to strong local fluctuations

at the pixel level. In all cases however the bi-orthogonal entropy is the smaller of them all. In table I we list the computed values for the “bears” image and the textures test image.

Table - Entropy values for test and “bears” images.

<i>Image</i>	<i>Bi-orthogonal</i>	<i>Histogram</i>	<i>Difference</i>
textures	0.1765	0.3572	0.5009
bears	0.0780	0.5642	0.3208

REFERENCES :

- [1] R.M. Haralick and L.G. Shapiro ; “Glossary of computer vision terms”, Pattern Recognition, **24** (1991) 69.
- [2] R.M. Haralick, K. Shanmugan and I. Dinstein ; “Texture features for image classification”, IEEE Trans. Systems, Man and Cybernetics, **3** (1973) 610.
- [3] A. Rosenfeld and E.B. Troy ; “Visual texture analysis”, Proc. UMR-Mervin, J. Kelly Communications Conference Section, **10-1**, 1970.
- [4] O.D. Fangeras and W.K. Pratt ; “Decorrelation methods of texture feature extraction”, IEEE Trans. Pattern Analysis and Machine Intelligence, **2** (1980) 323.
- [5] G.G. Lendaris and G.L. Stanley ; “Diffraction pattern sampling for automatic pattern recognition”, Proc. IEEE, **58** (1970) 198.
- [6] A. Rosenfeld ; “Automatic recognition of basic terrain types from aerial photographs”, Photogrammic Engineering, **28** (1962) 115.
- [7] B. Ashjari ; “Singular value decomposition texture measurements for image classification”, Thesis, Univ. Southern California, 1982.
- [8] M. Loève, “Probability theory”, Van Nostrand, 1955.
- [9] K. Karhunen, Ann. Acad. Sci. Fennicae, Ser. A1, Math. Phys., **37** (1946) 1.
- [10] J.S. Lim, “Two-dimensional signal and image processing”, Prentice-Hall, 1988.
- [11] T.D. Sanger, “Optimal unsupervised learning in feedforward neural networks”, MIT Artificial Intelligence Laboratory Tech. Report 1086, 1989.
- [12] T.D. Sanger, Neural Networks, **2** (1989) 459.
- [13] N. Aubry, R. Guygnet and R. Lima, J. Stat. Phys., **64** (1991) 683.
- [14] T. Pun, Signal Processing, **2** (1980) 223, **2** (1981) 210.
- [15] J.N. Kapur, P.K. Sahoo and A.K.C. Wong, Computer Graphics, Vision and Image Processing, **29** (1985) 273.

Figure Captions

- Fig. 1 (144×272) pixel image with uniform average gray level in 8×8 blocks.
- Fig. 2 Spectrum of the global ϕ_2 eigenfunction.
- Fig. 3 Entropy image for non-overlapping 8×8 blocks
- Fig. 4 Contours of the entropy image using non-overlapping blocks of size 8×8 .
- Fig. 5 Entropy image using blocks of size 8×8 separated by 4 pixels.
- Fig. 6 Contours of the entropy image shown in the figure 5.
- Fig. 7 Test image with two illumination levels.
- Fig. 8 Block entropy for the test image of figure 7.
- Fig. 9 Contours obtained from the entropy image in figure 8.
- Fig. 10 Bears. A real world test image (240×320) .
- Fig. 11 Spectrum of the global ϕ_2 eigenfunction in the “bears” image.
- Fig. 12 Spectrum of the global ϕ_3 eigenfunction in the “bears” image.
- Fig. 13 Spectrum of the global ϕ_{60} eigenfunction in the “bears” image.
- Fig. 14 “Bears” entropy image with blocks of size 18×18 separated by 4 pixels.
- Fig. 15 “Bears” entropy image with blocks of size 4×4 separated by 1 pixel.
- Fig. 16 Three-dimensional 18×18 entropy image.
- Fig. 17 Three-dimensional 4×4 entropy image.
- Fig. 18 Contours obtained from the 4×4 entropy image.